CSE 587 – Data Intensive Computing

Project Report Phase – 3

Team Details:

Name: Saad Ahmed
UB Person # 50483867
Email: saadahm2@buffalo.edu

Name: Abdul Shahid Mohammed
UB Person # 50465629
Email: am422@buffalo.edu

Project Title: Exploring the Relationship between Health and Lifestyle Factors and Depression

Problem Statement:

The goal of this project is to identify the key health and lifestyle factors that are associated with an increased risk of depression among adults. Using data from the NHANES health survey (United States Department of Health and Human Services, 2012), we will investigate how factors such as age, weight, drug use, alcohol consumption, diabetes, and marital status are related to the likelihood of experiencing depression. By analysing these relationships, we hope to gain insights into the underlying causes of depression and develop strategies for preventing and treating this common mental health condition.

Things covered in phase-1:

- Data cleaning
- Exploratory Data Analysis

Things covered in phase 2:

- Implementation of machine learning models
- Learning from the models

Selection of ML model for application:

We have implemented 5 ML models in phase 2 which are Logistic regression, Decision trees, KNN classifier, Naïve Bayes Classier, and State Vector Machines. Among the models implemented, we have chosen to implement Logistic regression in the final application based on the evaluation results of each model. The models are evaluated on the metrics such as accuracy, precision and recall using the classification report.

Model	Precision		Recall		f1-score		Accuracy
	0	1	0	1	0	1	
Logistic Regression	0.75	0.55	0.96	0.45	0.85	0.42	0.75
Decision Tree	0.75	0.41	0.92	0.15	0.83	0.22	0.72
SVM	0.75	0.34	0.85	0.21	0.8	0.26	0.68
KNN	0.76	0.5	0.97	0.11	0.85	0.15	0.75
Naïve Bayes	0.79	0.38	0.8	0.38	0.8	0.38	0.69

The table above shows the values obtained for different models we have trained on the data. Among the models, Logistic regression has performed well on the test data set to classify the people with and without depression. The precision, recall and f1-score values are comparatively higher for both the classes 0 and 1 for Logistic regression. Hence, we have decided to implement the product using this model.

Features of Web Application:

We chose to build a Flask web application based on our model to make the product available to users, who can check their health status by giving the related input in the web page.

To build the API, we created the pickle file of the model which can be used in the API to make the predictions based on user data which are collected through the web form created using HTML and CSS. The user can input the details such as BMI, age, marital status, alcohol consumption, drug usage, family history, education, diabetes status, and poverty level. Based on the input features, the model predicts if the person is at risk of depression or not.

The application also provides recommendations to users based on their input to help them improve their life style which can help them to solve their problems. The application also generates a bar plot to show the impact of each input feature on the prediction. The plot shows the positive coefficients of the logistic regression model and their corresponding input features. The users can learn which feature impacts the risk of getting impacted with depression.

Building Flask Web API:

The `app.py` file has the main code using Flask's routing mechanism. The application define two routes: `'/'` and `'/predict'`. The `'/'` route renders the home page index.html while the `'/predict'` route processes the form data and generates the prediction and plot. The `'/results'` route displays the prediction output, input form data, and plot image.

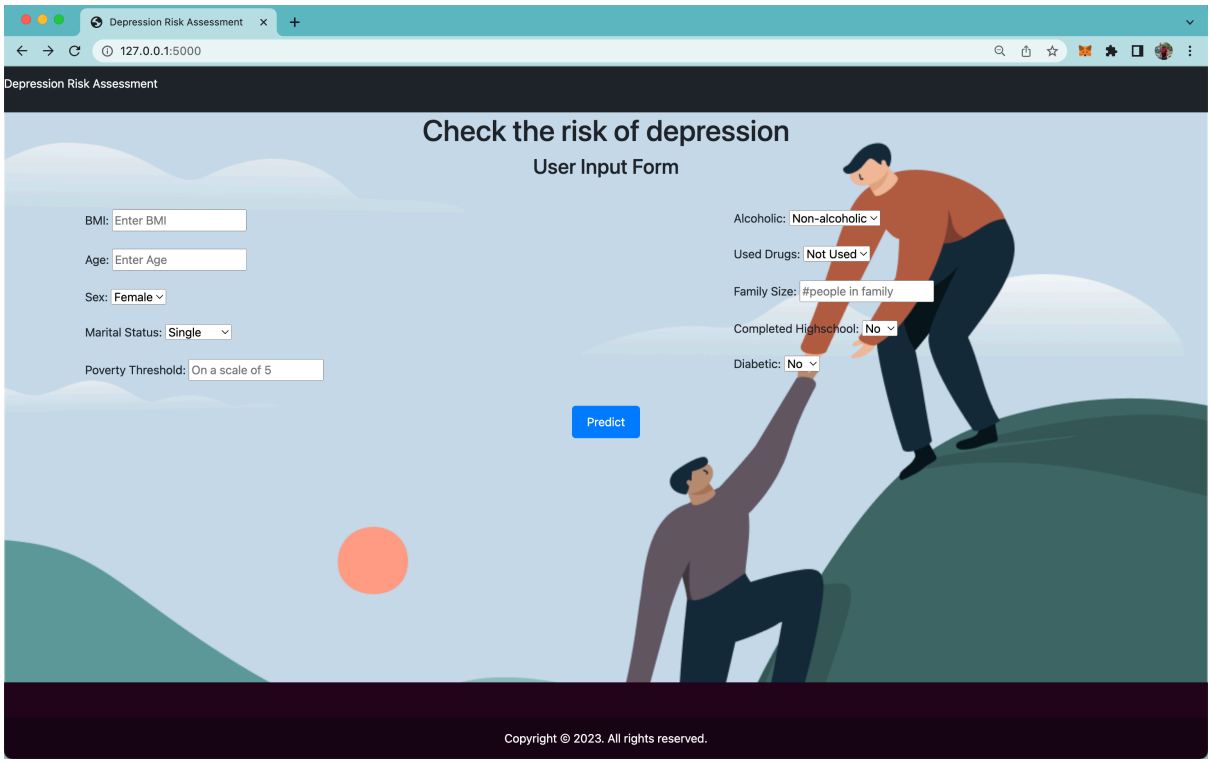
The code also has the custom logic to calculate some necessary input features based on the user input. For example, one of the input feature to the model is the health status (underweight, healthy, overweight, obese). This can be found out using the BMI value given as input in web form by user. Similarly, we also have logic to decode the financial status based on the poverty threshold given as input by user.

User Interface:

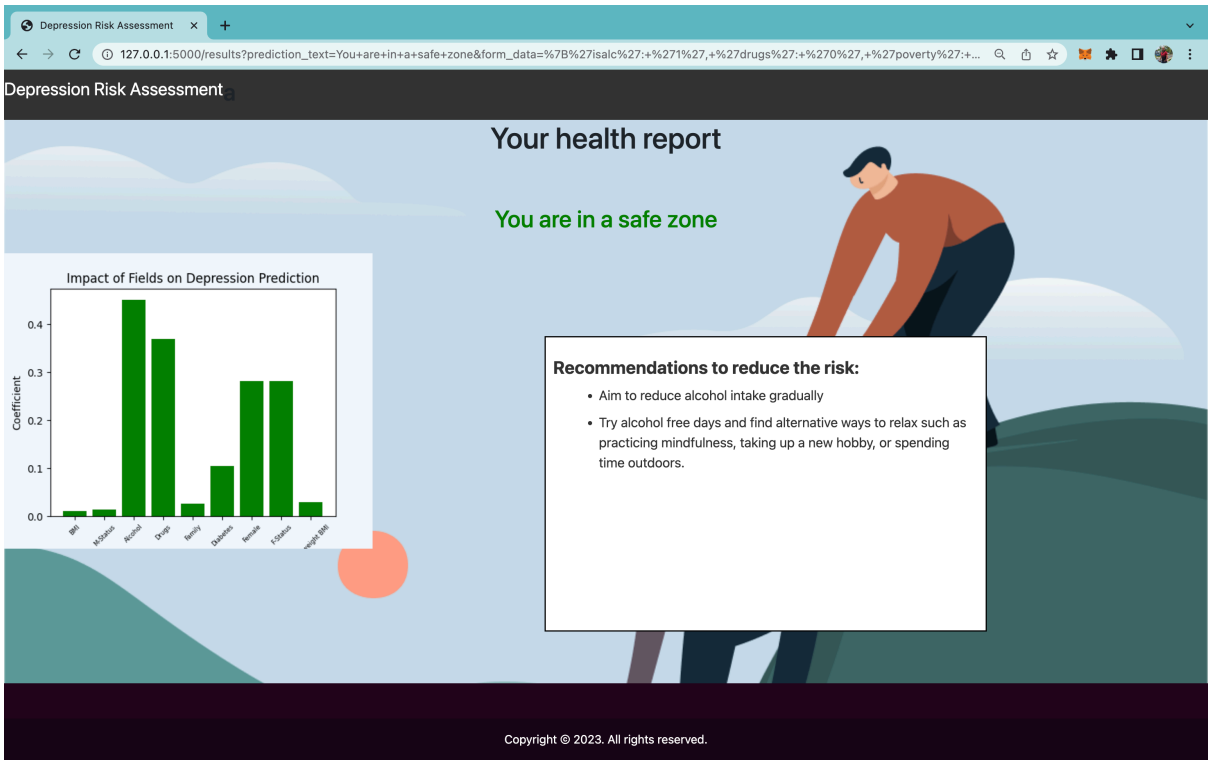
The user interface was built using HTML and CSS and it interacts with flask API to pass the parameters and display the results using POST and GET methods.

There are two web pages, one takes the input from user and allows the user to run the predictions on the input data. The results, recommendations and plots will be displayed on the second page which will be redirected upon clicking Predict button on the first page.

Below is the screenshot of user input form:



Below is the screenshot of results page:



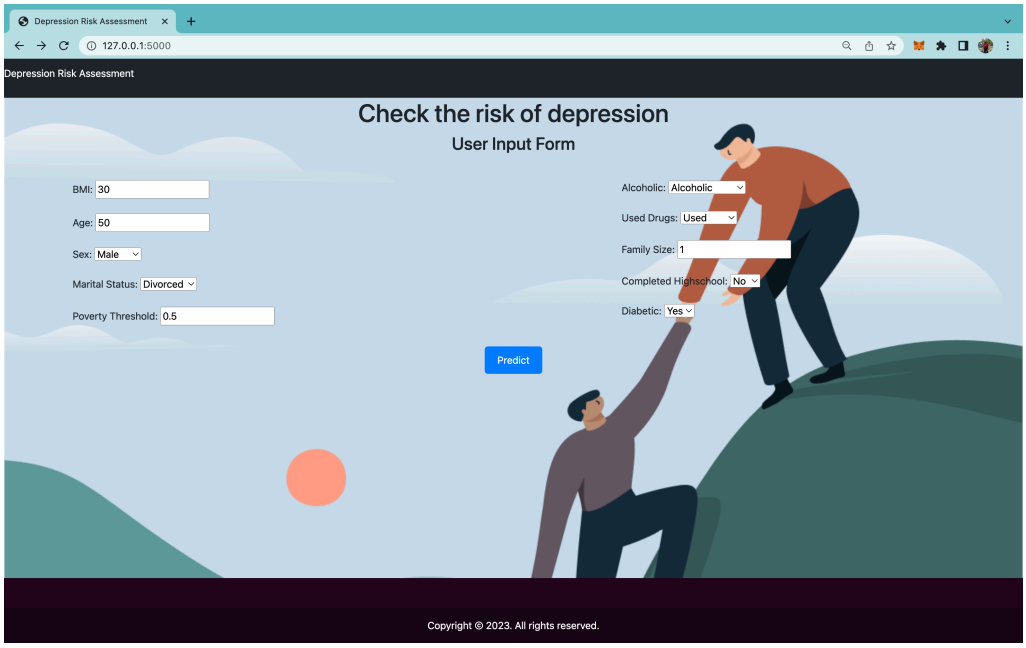
Demo – User Interaction, results, and recommendations:

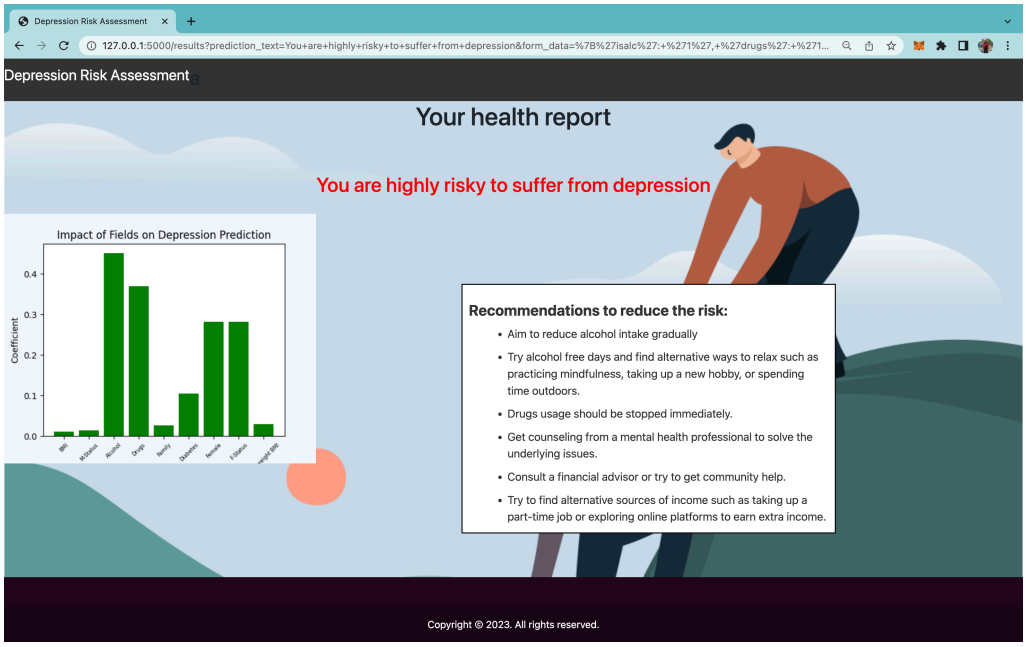
This section covers the demo of the web application by giving input to the user input form and by displaying the results.

There are two outputs of the application, One is where the model predicts the output as '0' i.e., user is not in the risk of depression based on his lifestyle and the other is where the user is in risky. The recommendations will be given out in either case, if the user needs based on his inputs.

Case 1: User is in the risk of depression

The below screenshot demonstrates the user input as below:



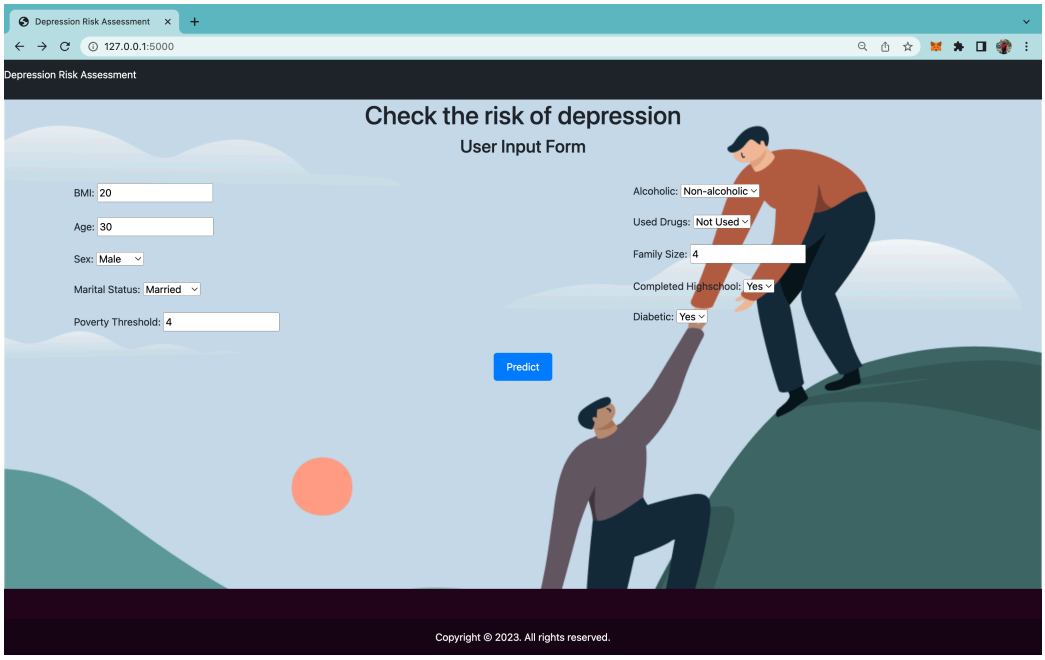


We can see the output of the model on the results page which says the user is risky to suffer from depression. Also the page displays the corresponding recommendations based on the user input.

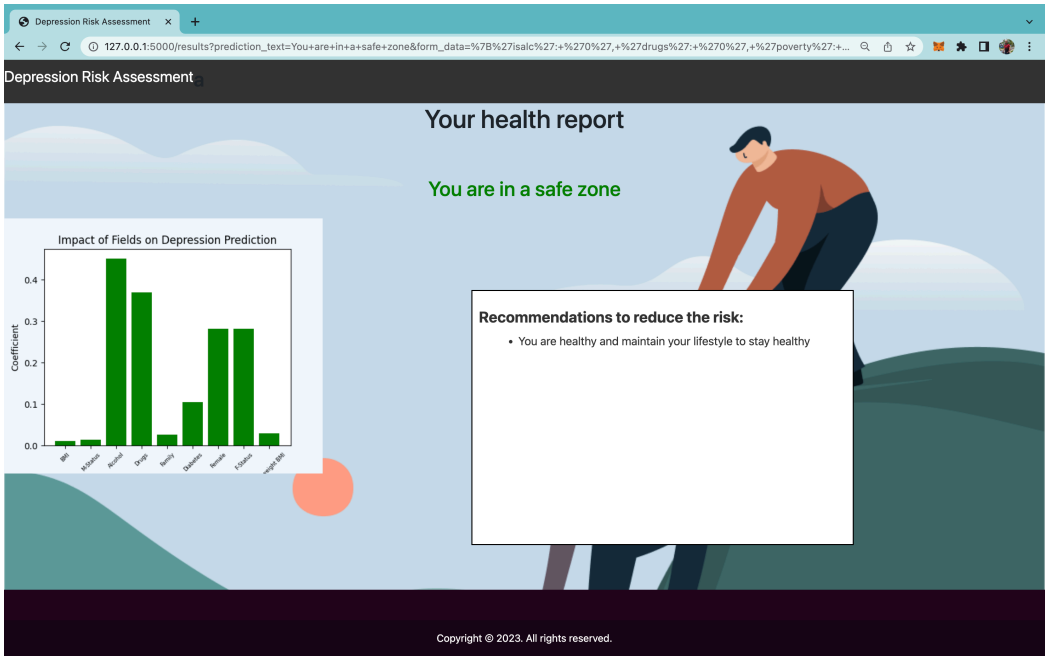
The page also has a plot, which shows the impact of each field on depression prediction.

Case 2: User is not in the risk of depression

Below screenshot shows the input fields filled with details of another user who is maintain a healthy lifestyle and who could be not in the risk of depression.



The results page displays that 'You are in safe zone' which means that user has a good lifestyle and the recommendations show that he should maintain the lifestyle.



Instructions to setup and run the project:

1. Download the src folder and follow the below steps.
2. In windows OS, Create a virtual environment using command “python -m venv env_name”. If you are using mac, follow the corresponding commands to create environment.
3. Activate the environment using the command “<env_name>\Scripts\activate.bat” in Windows.
4. Python version should be 3.10 and install all the dependencies mentioned in requirements.txt file using the command “pip install -r requirements.txt” or install libraries individually as required.
5. After the installation of libraries, Run app.py file (python app.py), you should be able to see the localhost URL in the terminal where you launched it.
6. Access the URL in any web browser and you should be able to see the user input form.
7. Input the data and click the predict button to see the results

Ideas to extend the project:

- **Idea 1:** The model can be trained on more data which can help to overcome the biases on certain features or inputs. This would help the model to perform better and can help to gain realistic recommendations to the users which can help them to better their lifestyle and stay healthy.
- **Idea 2:** Depression is a prevalent problem affecting individuals across all age groups. To identify and provide necessary support to students who may be susceptible to depression, a university survey can be conducted on different aspects of the education system and can train a model based on those features. This would enable the identification of students who require assistance to overcome depression among the students and the university can also find out the problematic things in education system.